Adaptive Differential Evolution Applied to Point Matching 2D GIS Data

Noel Khan * Ferrante Neri *† Samad Ahmadi *
* School of Computer Science and Informatics, De Montfort University,
The Gateway, Leicester LE1 9B1L, United Kingdom
Email: noelkhan@verizon.net, fneri@dmu.ac.uk, sahmadi@dmu.ac.uk
† Department of Mathematical Information Technology, P.O. Box 35 (Agora),
40014 University of Jyväskylä, Finland

Abstract—**The impetus behind data analytics and integration is the need for greater insight and data visibility, but since a growing share of our data is multimedia, there is a parallel need for methods that can align multimedia data. This paper explores georeferencing, which is used to combine spatial datasets and used here to align map images to 2D GIS models. This paper surveys various approaches for building the key components of a georeferencing solution, notes their strengths and weaknesses, and comments on their trajectory to help orient future work. The implementation presented here uses Hough transforms for feature detection, nearest neighbor correspondences with simplistic similarity measures, and a population based optimizer. The comparison among metaheuristics has shown that Differential Evolution (DE) frameworks appear especially suited for this problem. In particular, the controlled randomization of DE parameters appears to display the best performance in terms of execution time and competitive performance in terms of function evaluations even with respect to more complex memetic implementations.**

I. Introduction

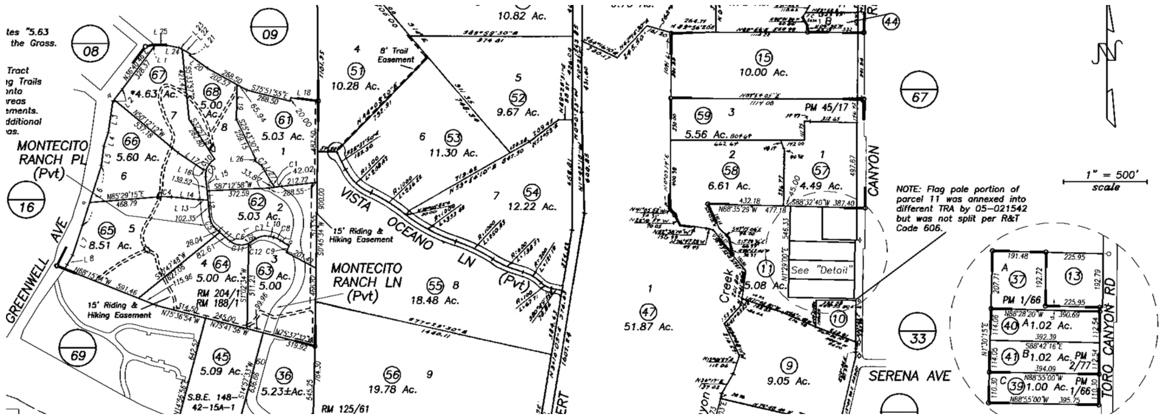
One of the main functions of government is managing land under their authority, which is supported by producing maps that portray their geography in a variety of ways. Maps from satellite imagery depict the real world as it existed at a point in time, cadastral maps show property boundaries based on official deeds, and tax maps aggregate or partition land into wide area or individual taxable units. Each of these representations are different views of the same geography, each view is fit for a particular purpose, and there are legitimate variations between them. Just as overlaying street maps on satellite imagery produce a composite with greater informational value than the individual views, the integration of other maps could likewise amplify the value of individual datasets and their aggregates. Here we overlay tax maps onto cadastral maps to depict the the many to many relationships between cadastral and tax boundaries.

In county level governments throughout the United states, a cadastral map is typically maintained as a seamless digital model of a geography using a geographic information system (GIS), whereas a paper based indexing system is used to manage a collection of tiled tax maps where each map contains annotations that reference adjacent maps. Tiled maps are often produced by scanning and "photo-shopping" maps submitted by private land developers, which yields digital images in tagged image file format (TIFF). A TIFF file is like a container for both images and meta-data called TIFF tags that describe, for example, image dimensions, palette, and compression, which tell rendering software how to process and display their content. The TIFF specification allows additional tags to be defined and published to support interoperability and the GeoTiff specification [1] defined a set of TIFF tags for storing geographic information such as GPS coordinates and coordinate spaces. Absent such geographic meta-data, images would be rendered either at the canvas origin or within a user defined rectangle, but geotiffs provide rendering software with enough information to render the image in a more geographically appropriate place by transforming the image's coordinate space to some other coordinate space.

A common coordinate space is a prerequisite for integrating geographical data, but image and model coordinate spaces are defined differently. The origin in image space is defined as the top-left corner so the y-axis increases downward and the extents of image space are bounded by zero and the width and height of the image in unit pixels. The origin in GIS model space is in the lower-left corner and often assigned non-zero coordinate values while the extents are arbitrary and defined by a standards setting body. Rendering software converts image coordinates to model coordinates using a transformation matrix that encodes translation, rotation, and scale. Differences in how the origin is defined is handled by the translation component of the transformation matrix, extents are handled by scaling, and the different y-axis convention is handled by a reflection matrix. Since the mechanism for transforming image coordinates to model coordinates is known, automatic georeferencing is essentially a parameter identification problem, which involves the search for the appropriate translation, rotation, and scaling parameters required to position a tiled map over its corresponding geography in the GIS.

Information on some types of maps do not necessarily exist on other maps, which means that only a subset of significant features can be correlated between geospatial datasets. The identification of corresponding points of interest, or features, is often a manual process, but properties of both GIS and

978-1-4799-7560-0/15 $31.00 © 2015 IEEE
DOI 10.1109/SSCI.2015.241



(a)

(b)

Fig. 1: Example input data. Figure (a) shows a property tax map with detailed inset (lower right corner) and leader lines (top right of map). Figure (b) shows the corresponding geography in the seamless GIS model

tax maps lend themselves to automatic feature identification. Using standard GIS tools, the location of each property could be abstracted to a coordinate by computing the centroid of each property boundary or, if the GIS contains labels such as each property's street address, then coordinates for those labels could be used to represent property locations. Tax maps uniquely identify taxable units using identifiers composed of a map's book and page numbers and block and lot numbers ("APN"). As shown in Figure 1a, lot numbers are enclosed in small circles of a standard diameter and are therefore easier to identify and take coordinates from than finding centroids of arbitrary shapes. More generally, therefore, automatic georeferencing is a point matching problem with an embedded parameter estimation problem.

Georeferencing is nontrivial due to the noise within and between the datasets. The different views of property (taxable parts vs. cadastral lines of ownership) constitute the most significant source of noise between the datasets, while a minor source of noise is imparted by differences in mapping conventions. For example, the tax maps may place property numbers dead center, whereas cadastral maps may place property numbers at a fixed perpendicular distance from the road to indicate the location of the property's entrance. Also, since tax maps are drawn at a fixed scale, there is the occasional need to draw detailed insets or to use leader-lines that point into a smaller part of the map. Finally, even though the creation of tax and cadastral maps are document driven, differences in which documents are used to update each type of map can contribute to variations.

This paper surveys the popular ways in which features are extracted from images, how relationships between features within each dataset are represented, how features between the datasets are correlated, and how the parameters of the transformation from image coordinates to model coordinates are estimated. The combination of feature extraction technique, knowledge representation, correlation approach, and optimization techniques are required to completely specify a georeferencing solution and this research includes the results of one such system.

The remainder of this paper is organized as follows. The literature review in Section II provides a broad overview of the application areas, techniques, and challenges. Section III offers a general discussion on key design decisions and Section IV goes over the experiments in this study. The paper ends with a summary of results found in Section V and conclusions in Section VI.

II. Literature Review

A. Applications

Automatic referencing lies at the intersection of optimization and computer vision, which is a subset of image processing. Registration is a special case of referencing that is limited to integrating datasets of the same type, e.g., two images. Popular registration applications include image mosaicing to create panoramas [4], [6]–[8], canceling out the effect of camera vibrations, building 3D models from stereo images [8]–[13], motion detection, and object tracking [7], [14], [15]. Common referencing applications include combining range and satellite imagery to create digital elevation models or local world models for robot navigation [16], or combining medical data acquired from ultrasounds, X-rays or MRIs [15], [17], [18]. Other applications include pattern [9], shape [9], or facial recognition [11].

B. Challenges

Fundamentally, the challenges referencing regard the quality and quantity of input and the tradeoff between computational speed and accuracy. Factors affecting input include variations in data production or acquisition [13], distortions [18], blurring [18], occlusion [19], and variations in quantity such as extraneous data or noise [12], [20], missing data [14], outliers [20], or variations in the density or distribution of data [9]. These input related issues will hereinafter be referred to as noise and are addressed through greater levels of data abstraction as will be discussed later.

Since transformations and similarity measures are described mathematically, analytic methods involving derivatives of those expressions can be used to compute solutions. Among the fastest referencing techniques use analytic methods to compute the exact transform of noiseless datasets [21], but since real world problems are noisy, most techniques involve searching for solutions using regression or population based meta-heuristics. Generally, regression works relatively well for small 2D datasets with small local changes and little overall noise, whereas meta-heuristics also work well in large and noisy ND datasets. Self adaptation typically increases dimensionality [22], [23], which thereby decreases the feasibility of regression.

Both techniques contain internal parametrization problems. Since they are heavily influenced by starting point (initial pose) regression techniques are often fooled into local minima and, to compensate, researchers proposed various hierarchical techniques that parametrized depth, changes in resolution, and the number of iterations [7], [11], [12], [16]. Common population based parameters include computational budget, thresholds, and population size. These parameters can have a significant impact on an implementation's accuracy, control, and capability.

Much of the applied research in referencing contain literature reviews followed by a litany of reasons why existing implementations would not work well for the particular application at hand. Such observations can be explained by the No Free Lunch Theorem, which represents an algorithm's performance as some unit distribution over all problems and states that if an algorithm performs well for a particular application then it necessary performs poorly in others [23]. In other words, the average performance of any algorithm across all problems will be no better than random; broad based algorithms will perform worse than application specific algorithms. The specificity and quantity of algorithms, however, can be mitigated through the use of memetic algorithms containing a variety of search operators [22].

C. Knowledge Representation

Barring abstraction, every pixel in an image would be a feature that would have to be matched to a point in model space, but such a detailed representation of the image would make referencing super sensitive to annotations, noise, color, illumination, and image quality. Extracting particular points from an image encodes some knowledge about the source data and thereby yields an abstraction of the original.

What other knowledge can be extracted and how should it be represented? Since an image is a 2D array of pixel values, some researchers proposed extracting the gradient of neighboring intensity values to represent the image as a force field diagram, while others extended that idea by binning the image into larger pseudo-pixels with averaged pixel values in order to further abstract away detail [24]. Other researchers extracted points of interest and then created orthogonal histograms of the entire image space [25], histograms projected onto one dimension [26], or radial histograms limited to the immediate neighborhood of extracted points [27]. Yet others encoded sets of points using minimum spanning trees [27] or B-splines [4], [20], [28], [29], both of which yield more detail about the distances and directions to neighboring points. Throughout the literature, less abstract or image-based operations work better locally, but more abstract feature-based operations are better globally. The tradeoff, therefore, is between the scope of the search and abstraction, which removes detail (required for local discrimination) as well as noise. As shown in Figure 2, the general trajectory of research and development is moving towards greater abstraction to address noise and towards a greater consideration of local neighborhoods to address quality.

Note that gradient based methods examine changes in pixel intensity to detect corners and edges, while the examination of the rate of change is used to detect more abstract features like ridges and areas. In either case, the roots of the first and second derivatives of the image yield concise self-describing information about significant points of interest [30] and the consideration of each point's relationship to other points, i.e., local neighborhood, similarly enriches the description of each point. In other words, the underlying trend is actually the increased use of meta-data, which in turn increases the expressiveness of knowledge.

D. Correlation

The preponderance of evidence in literature suggests that referencing quality is directly proportional to expressiveness. That is, the ability to discriminate, discern, and correlate depends on the amount of detail perceived. A human can instantly pick out corresponding points on a pair of stereo images, but a computer's view is limited only to the detail extracted or derived from the images.

Using only coordinate information, points from one dataset can be correlated to points in another by associating each point to its nearest neighbor in the other set, but such an approach is highly susceptible to both incorrect and non-unique N:1 correspondences. However, if edges between points in each set were known, then correlation could be based on matching the lengths and angles of edges leaving each point (a.k.a. Shape Context [27]). Likewise, any increase in the amount of meta-data used to describe each point also increases the likelihood of achieving correct and unique 1:1 correspondences.

This is reflected in the dominant point matching algorithms: the Iterative Closest Point (ICP) algorithm uses nearest neighbor correspondences, which makes it highly susceptible to noise and initial pose whereas the Robust Point Matching (RPM) algorithm correlates all point combinations between sets (star correspondences), which broadens the domain of convergence but yields "compromise solutions" [31] as it applies equal weight to legitimate points and noise. [32] reported star correspondences worked well for rigid body transformations, but was unable to correlate points in the presence of noise even when both point sets contained an equal number of points.

Since real world problems are often plagued with noise, even the most expressive feature descriptions would be derailed if similarity was based on matching exact values. There are two schools of thought delineated on the cardinality of correspondences: one school accepts N:1 correspondences but selects the most likely one based on a statistic, while the other achieves 1:1 correspondences by leveraging derivatives to identify the best match. Statistics based techniques include Gaussian weighted multiple matches [20], [27], bi-partite graph matching [27], Bayesian inference [20], soft assign [10], and mutual information [17], [33]. Statistical methods have also been used to gauge correspondence quality and reduce the influence of noise, but are admittedly computationally intensive and have limits on scalability [31]. Derivative based techniques include: M-estimators [27], maximum votes [6], maximum likelihood [9], minimum shape context [27], and minimal euclidean distance [20], [21].

The final decision regarding correlation is how to compare quantized properties of features. A single measure of how well one dataset fits to another is customarily achieved by summing differences. However, since some differences may be negative and others positive, the net effect could be canceled out by summation [34], which is why the differences are typically squared. Alternatives to the sum of squared differences (SSD) include root mean square (RMS), the sum of absolute differences (SAD), and the mean of absolute differences (MAD).

E. Optimization

Exact methods guarantee that an optimal solution will converge to the optimal solution if this solution exists, whereas metaheuristics offer no such guarantee since they operate on an abstracted or estimated view of the problem. Metaheuristics are the only option in most real world cases when for example the analytical formulation of the objective function is not available or even when real world limitations such as uncertainties with unknown features are present within the problem.

Among metaheuristics, population based algorithms represent a parallel generate and test strategy where members of the population encode solutions of varying quality, which are scored by an objective function, see [35]. Members of the population are perturbed using some combination of the following operators: crossover, which usually combines in some ways members of the population, or mutation, which produces randomized parameter changes within the solution. These operators can be understood as operators of different nature as the crossover exploits the existing pieces of information stored within the population while the mutation generates perturbs the existing search directions and searches for new ones by exploring untested areas of the search space. Some algorithms select and perturb higher scoring individuals in hopes of producing offspring with some combination of favourable characteristics found in the parents, while other algorithms perturb combinations of all members to produce a surplus population that is then truncated down to the higher performing individuals. Classical population based algorithms, which are all thoroughly described in [22] [23] [36], include: genetic algorithms (GA), evolutionary strategy (ES), particle swarms (PSO) [37], and differential evolution (DE).

This study tests and compare these classical metaheuristics as well as some more modern algorithmic implementations. The main aim is to clarify relative advantages and disadvantages of each optimizer for the data science problem under investigation. Although the features of the main metaheuristic frameworks are well-known, let us briefly summarise and highlight what, in the view of the authors, the main differences among them are.

The main focus of the search in GAs is driven by crossover, which yields some interpolation of genes between a selected pair of solutions [23]. On the contrary, in ES the search of new promising solutions is mostly driven by mutation from which the top performing solutions are picked out of an expanded population to seed the subsequent generation [23]. Both these frameworks have selection/replacement schemes that take into account the entire population individuals (along with their fitness) to select the most promising search directions.

Swarm intelligence algorithms and DE are characterised by a different replacement logic. In PSO, members of the population are not replaced, but rather their position is updated

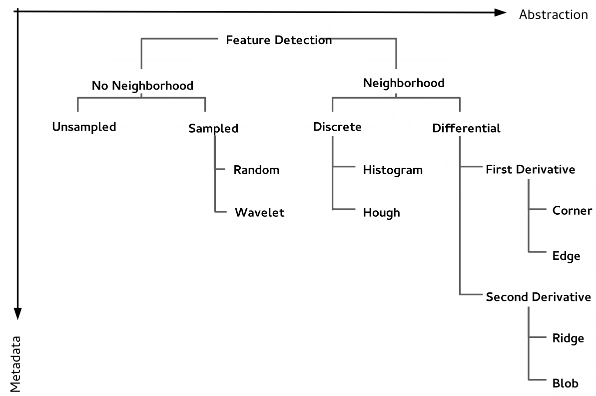


Fig. 2: Progression of feature detection techniques

using a weighted vector sum of three terms. Several perturbation logics exist. If a particle's new position outperforms its prior or global best, then memory of those positions are updated, which change the direction of the individual particles and overall movement of the swarm. Exploitation is driven by linear decreases in inertia each iteration, which transitions the search from global to local. DE's mutation uses the difference between two vectors and makes use of some randomisation to generate a new solution [36]. Despite its simplicity, DE contains an implicit self-adaptation [38] as the average search radius progressively shrinks during optimization and thereby enhances exploitation. The main feature common to swarm intelligence and DE is that the pairwise comparison and replacement logic is performed in one-to-one spawning fashion: the update occurs when the newly generated point/position outperforms the solution that generated it.

III. Design

A georeferencing solution is completely specified upon selecting a particular feature extraction technique, knowledge representation, correlation approach, and optimization technique, but in real world applications these choices are often constrained by nontechnical considerations such as project schedule or nonfunctional requirements such as performance. Orange County (CA), for example, has manually georeferenced about 60,000 survey-map images in the past 15 years at an average rate of 2 images per working hour, so a reasonably compelling lower bound for system performance may be 20 images per hour. At that rate, it would take 5 weeks around the clock to georeference their 16,000 tax maps and this performance requirement limits the full range of options available to each system component. Scalability is another key nonfunctional requirement since the road map for the present work includes use in 3D geospatial datasets as well as arbitrary map images rather than only tax maps.

Accordingly, metaheuristic optimization appeared to be the more promising optimization approach since it better accommodates higher dimensional problems and is less dependent on initial pose, which avoids multiple runs per problem and therefore longer processing times. Regarding feature detection technique, OpenCV's Hough circle transform was used to detect circles around lot numbers in under 30 seconds, but imposed a filesize limitation since the algorithm is memory intensive. This was an acceptable compromise for the current application since nearly all tax map images were under that 400kb limit. Property boundaries in both datasets were represented only as a set of coordinates since the relationships between points within each set, such as the distances and angles between them, were expected to overly constrain the problem if there were different numbers of points or high dissimilarity between the two datasets. Experimentation, therefore, centered around identifying the best fit metaheuristic optimization algorithm, correlation approach, and objective function.

There were two simplifying assumptions used in the present work. Tax maps were uniformly scaled, which meant $S_x = S_y$ and this observation reduced the dimensionality of the problem from 5D to 4D. Secondly, since tax map images are named after their book and page numbers and since property records in the GIS have an APN attribute, it was possible to prune model state space by selecting records where part of the APN attribute from the GIS matched the image filename.

IV. Experiment

This summer project afforded only enough time to test a couple variations of correlation approaches, objective functions, and metaheuristic algorithms. To reduce the combinations of options within those three components, a baseline objective function was established first so that the best metaheuristic algorithm could be selected and factored out from subsequent tests. A generate and test strategy compared results from a baseline system to results from alternatives. The selection criteria for a particular system specification was based on the percentage of images that were accurately and precisely georeferenced. Alternatives were vetted using a set of 4 problems to facilitate multiple test cycles and to identify variations that broadened the scope of solvable problems using a pair of problems that were correctly georeferenced by the baseline and a pair that were not. Successful variants were then run against another set of 36 problems to gauge improvements in accuracy and precision, which were assessed qualitatively and ranked. The best variant was run unattended against a set of nearly 20,000 tax maps from Santa Barbara and Orange Counties (CA). To assess solution quality, final variances logged during processing were mined to compute average differences and standard deviations between measures. For a more detailed analysis, 200 results within the Orage County cities of La Palma and San Clemente were quantitatively evaluated to gauge georeferencing quality in terms of translation, scale, and rotation errors.

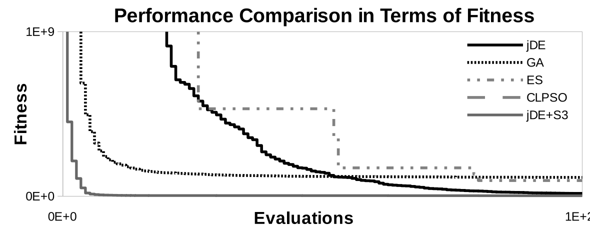
A. Objective Functions

Relatively simple measures were used because increases in expressiveness meant an increased computational burden to correlate points and would thereby have a disproportionate impact on performance. Accordingly, the baseline objective function used the nearest neighbor correspondences and assessed the quality of the solution using scaled sums of absolute differences (SAD) between five measures, which included the distances d_i between nearest neighbors and four meta-data based measures. Since nearest neighbor correspondences are not 1:1, the distinct subset of matched points with minimal distances were scored as d while all other trial points were assigned a penalty score d' equal to $100 * max(\mathbf{d})$, both of which allowed the optimization of translation parameters t_x and t_y. Likewise, every other characteristic desirable in the solution requires a measure so it can be optimized. Scale was optimized by comparing the height MBR_H and width MBR_W of the minimum bounding rectangles (MBR) around both datasets, which implicitly encoded information to distinguish orthogonal rotations. Additional detail about rotation was derived from the length δ of the line from the top-left corner of the MBR to the left-most point. The absolute differences of these meta-data derived terms were scaled by constants c_i, which imparted a form of hierarchical processing since more highly scaled parameters were estimated first. Two alternative objective functions were variations on the baseline where one used SSD rather than SAD to aggregate measures and the other used star correspondences rather than nearest neighborbor correspondences. The fourth and final objective function used vertical and horizontal historgrams to match the point sets.

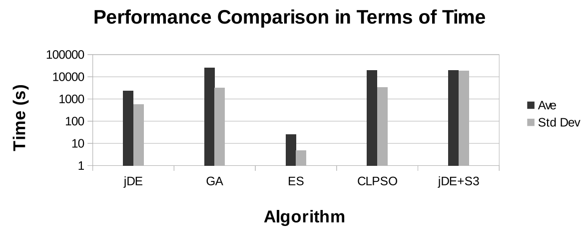
B. Metaheuristic Algorithms

Using a representative cross section of metaheuristic algorithms, the selection criteria for the best algorithm was the one that exhibited the best accuracy and performance in terms of execution time. Each algorithm was tested against the same set of 10 problems, one run per problem, random starting points, a 50 member population, a $1e^{+6}$ evaluation computational budget, a diversity-based exit criteria set to $1e^{-9}$. Algorithms included a real-valued GA with tournament parent selection, box child selection, Gaussian mutation, and generational replacement [39]. ES was implemented with two parent recombination, unbiased parent and gene selection, Gaussian mutation, $1/5$ success rule, and survivor selection based on the comma strategy see [39]. CLPSO followed its original implementation in [40]. An adaptive version of differential evolution called jDE [41] used rand/1 mutation and an exponential crossover, while a memetic form of jDE improved each trial solution using the S3 algorithm proposed in [42]. It must be remarked that in this study we have chosen to use some examples of classical metaheuristic and some relatively modern implementations (CLPSO and jDE) which are these days renown as simple and efficient algorithmic frameworks. The memetic implementation can be seen as an attempt to integrate local search within an efficient algorithm.

The parameter settings used for the algorithms are as follows: (GA) tournament size equal to 2, a σ within the mutation of 0.5. (ES) initial mutation step size $\sigma = 2$ and offspring population $\lambda = 7$. (CLPSO) maximum velocity 10, minimum and maximum inertia, 5 and 10 respectively, weight



(a)



(b)

Fig. 3: Meta-heuristic algorithm performance on the same set of 10 4D problems. Note the logarithmic scale for time in figure (b).

2, and constant $Pc = 0.5$, see [40]. (DE) scale factor and crossover rate equal to 0.5. (jDE) thresholds $\tau_1 - \tau_2 = 0.5$ and upper and lower bounds of scale factor 1 and 0.1 as indicated in [41]. (jDE+S3) as defined above with S3 using a large initial step-size $\delta = 5$.

V. Results

The choice of how measures are aggregated affected the shape and amplitude of problem landscapes and thereby affected search algorithm efficacy. Metrics captured during optimization indicate that the signal to noise ratios using SSD had better discriminatory power than SAD, but those effects did not broaden the scope of solvable problems. On the set of 36 problems, the SSD variant improved baseline results in 64% of problems, did worse on 33%, and was otherwise as wrong as the baseline. The other variants were abandoned upon failure to solve the 4 vetting problems possibly because of inadequately tuned parameters.

The metaheuristic algorithms that exhibited the best performance were jDE proposed in [41] and the jDE+S3 variant. As shown in Figure 3a the memetic algorithm with an embedded steepest descent local search operator within a jDE framework exhibited the best performance with respect to convergence speed and fitness evaluations, but had average execution times and standard deviations more than an order of magnitude greater than jDE alone as shown in Figure 3b.

jDE offered the best balance between speed and accuracy and was used in the large scale test of nearly 20,000 images, but could not meet the 5 week processing schedule without being distributed across 12 computing nodes. About 80% of the geography covered by the datasets were urban or suburban

TABLE I: Summary results from nearly $20,000$ images showing the variances of MBR-derived measures

	MBR_W ft	MBR_H ft	MBR_θ °
Average	515	502	6
Mean	16	17	1
StdDev	2467	3125	9
Min	0	0	0
Max	132548	146624	80

TABLE II: Detailed error metrics from a subset of results from suburban areas

	Rotation °	Placement ft	Scale %
Average	8	127	10
Mean	2	28	2
StdDev	24	276	23
Min	0	6	0
Max	180	1613	132

areas that had dense point sets and good results, but large scale or rural areas performed poorly likely because of the insufficient control imparted by fewer points. This suggests that larger point sets blur or average the differences between the sets. Generally, about 50% of the results had negligible variances, which indicate the inputs had high similarity and outputs had excellent alignment. An additional 35% to 45% of all images were within one standard deviation, but the magnitude of the variances shown in Table I suggest that either major failures in sparse areas comprising 20% of the geography contributed to large spreads or that the MBR-based scale measure was insufficiently weighted. A detailed analysis of suburban areas within La Palma and San Clemente yielded the statistics shown in Table II. Rotation errors appear more aesthetically egregious than scale or translation errors and 56% of the results were within 2.5° of the model and another 19% were within 5°. However, because these results were for a suburban area, these quantitative results do not incorporate results from large scale rural areas, which had significant placement errors.

VI. Conclusion

Point matching is a challenging problem because it often involves noisy datasets, large state spaces, and a modest number of parameters that need to be estimated to transform points in image space to points in model space. The single most important system component is arguably the computational framework. Although the baseline objective function employed here was simplistic, the system produced results in suburban and urban areas with accuracy and precision comparable to manually georeferenced images. It should be noted that there is value even when a map is roughly georeferenced, because being in the immediate vicinity of its correct position significantly reduces the amount of manual effort required to correct georeferencing. The meta-data measures evaluated each point set as a whole, which was resilient to noise but too abstract for precise parameter estimation. The histogram based objective function, by binning the coordinate space into 10 partitions along each axis, offered a more granular evaluation of the datasets, but was insufficient in itself to tune rotation and scale. Future work may investigate meta-data measures specific to each point that can then be used determine correspondence ratios between points in each set from which largely 1:1 correspondences can be determined. Furthermore, the baseline objective and its variants strongly supported the need to explicitly handle outliers to avoid assimilating their effect into the final solution.

References

[1] N. Ritter and M. Ruth, "Geotiff format specification," http://www.remotesensing.org/geotiff/spec/geotiffhome.html, 2000.

[2] R. Stephens, *Two-Dimensional Transformations in Visual Basic Graphics Programming*, 2nd ed. Wiley, 2000, pp. 363–416.

[3] G. Bradski and A. Kaehler, *Image Transforms in* Learning OpenCV. Sebastopol: OReilly, 2008.

[4] M. Xia and B. Liu, "Image registration by super-curves," *IEEE Transactions Image Processing*, pp. 720–732, 2004.

[5] A. Kadyrov and M. Petmu, "Fast registration for 2d images, the clock algorithm," in *Proc. Int. Conf. Image Processing*, vol. 2, 2003, pp. 715–718.

[6] Q. Wei, F. Zhizhong, L. Lingqiao, and D. Zaiqiang, "Image registration based on feature extraction and voting strategy," in *Int. Conf. Communications, Circuits and Systems*, 2008, pp. 706–709.

[7] Q. Zheng and R. Chellappa, "A computational vision approach to image registration," in *11th IAPR Int. Conf. Pattern Recognition*, 1992, pp. 193–197.

[8] G. Caner, M. Tekalp, G. Sharma, and W. Heinzelman, "An adaptive filtering framework for image registration," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, 2005, pp. 885–888.

[9] J. Ma, J. Zhao, J. Tian, Z. Tu, and A. L. Yuille, "Robust estimation of nonrigid transformation for point set registration," in *IEEE Conf. Computer Vision and Pattern Recognition*, 2013, pp. 2147–2154.

[10] P. Zhang, S. Wang, Y. Qiao, J. Yang, and Y. Gao, "Affine softassign with bidirectional distance for point matching", booktitle =."

[11] C. Li, J. Xue, S. Du, and N. Zheng, "A fast multi-resolution iterative closest point algorithm," in *Chinese Conf. Pattern Recognition*, 2010, pp. 1–5.

[12] C. Li, X. Luo, S. Du, and L. Xiao, "A method of registration based on skeleton for 2-d shapes," in *5th Int. Congress Image and Signal Processing*, 2012, pp. 810–813.

[13] D. Cobzag, H. Zhang, and M. Jagersand, "A comparative analysis of geometric and image-based volumetric and intensity data registration algorithms," in *Proc. 2002 IEEE Int. Conf. Robotics and Automation*, 2002, pp. 2506–2511.

[14] R. Sandhu, S. Dambreville, and A. Tannenbaum, in *IEEE Conf. Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[15] I. Sahin, "Registration of 2d point sets by complex translation and rotation operations," in *32nd Annu. Int. Conf. IEEE EMBS*, 2010, pp. 5605–5607.

[16] X. Zhang, L. Li, and D. Tu, "Point cloud registration with 2d and 3d fusion information on mobile robot integrated vision system," in *Proceeding IEEE Int. Conf. Robotics and Biomimetics*, 2013, pp. 11 187–1192.

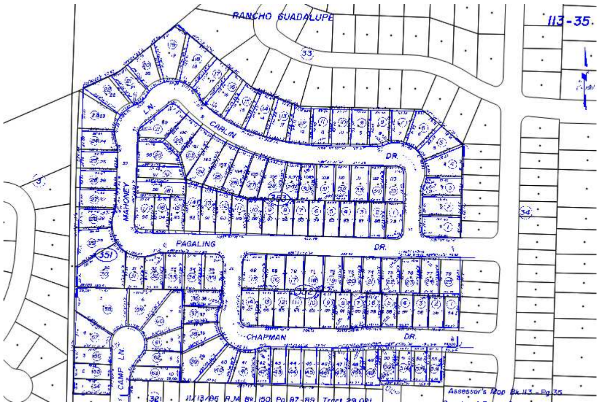
[17] M. N. Haque, M. R. Pickering, M. Biswas, M. R. Frater, J. M. Scarvell, and t. . P. N. Smith.

[18] M. S. Yasein and P. Agathoklis, "A feature-based image registration technique for images of different scale," in *IEEE Int. Symp. Circuits and Systems*, 2008, pp. 3558–3561.

[19] K. V. Arya, "An efficient 2d image registration algorithm," in *4th Int. Conf. Industrial and Information Systems*, 2009, pp. 354–357.

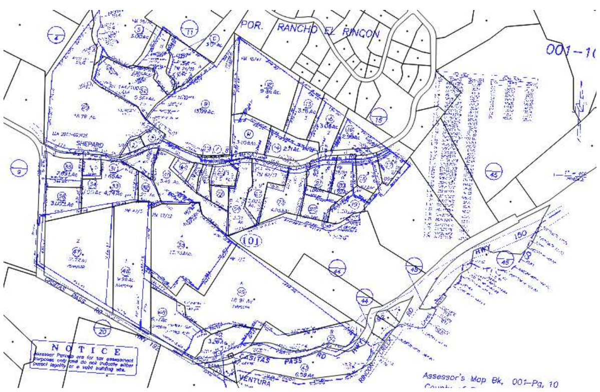
[20] H. Chui and A. Rangarajan, "A feature registration framework using mixture models," in *Proc. IEEE Workshop Mathematical Methods In Biomedical Image Analysis*, 2000, pp. 190–197.

[21] J. Ho, M. Yang, A. Rangarajan, and B. Vemuri, "A new affine registration algorithm for matching 2d point sets," in *IEEE Workshop Applications Of Computer Vision*, 2007, p. 25.



(a) Small scale suburban area with negligible rotation

(b) Medium scale suburban area with negligible rotation



(c) Large scale rural area with small rotation

(d) Medium scale urban area with large rotation

Fig. 4

[22] F. Neri, C. Cotta, and P. Moscato, *Handbook of Memetic Algorithms*, ser. Studies in Computational Intelligence. Springer, 2012, vol. 379.

[23] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*, ser. Natural Computing. Springer, 2003.

[24] "Edge detection," available at http://en.wikipedia.org/wiki/Edge_detection, 2015.

[25] M. Weinmann and B. Jutzi, "Fast and accurate point cloud registration by exploiting inverse cumulative histograms," in *Joint Urban Remote Sensing Event*, 2013, pp. 218–221.

[26] V. Daum, D. Hahn, and J. Hornegger, "A nonlinear projection scheme for fast rigid registration," in *IEEE Nuclear Science Symp. Conf. Record*, 2007, pp. 4022–4026.

[27] M. Rogers and J. Graham, "Robust and accurate registration of 2-d electrophoresis gels using point-matching," *IEEE Transactions Image Processing*, pp. 624–625, 2007.

[28] X. Wang and D. Feng, "Automatic hybrid registration for 2-dimensional ct abdominal images," in *IEEE 1st Symp. Multi-Agent Security and Survivability*, 2004, pp. 208–211.

[29] M. Hansen, R. Larsen, B. Glocker, and N. Navab, "Adaptive parametrization of multivariate b-splines for image registration," in *IEEE Conf. Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[30] "Ridge detection," available at http://en.wikipedia.org/wiki/Ridge_detection, 2015.

[31] M. Sofka, G. Yang, and C. V. Stewart, "Simultaneous covariance driven correspondence (cdc) and transformation estimation in the expectation maximization framework," in *IEEE Conf. Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[32] J. Xu, J. Hu, and X. Jia, "Genetic algorithm for distorted point set matching," in *6th Int. Congress Image and Signal Processing*, 2013, pp. 1724–1729.

[33] L. Xu, J. Liu, W. Zhan, and L. Gu, "A novel algorithm for ct-ultrasound registration," in *IEEE Point-Of-Care Healthcare Technologies*, 2013, pp. 101–104.

[34] S. Brown, "Least squares - the gory details," available at http://brownmath.com/stat/leastsq.htm, 2007.

[35] A. Prügel-Bennett, "Benefits of a Population: Five Mechanisms That Advantage Population-Based Algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 4, pp. 500–517, 2010.

[36] K. V. Price, R. Storn, and J. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*. Springer, 2005.

[37] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of IEEE International Conference on Neural Networks*, 1995, pp. 1942–1948.

[38] V. Feoktistov, *Differential Evolution in Search of Solutions*. Springer, 2006, pp. 83–86.

[39] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computation*. Berlin: Springer Verlag, 2003, pp. 175–188.

[40] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 281–295, 2006.

[41] J. Brest, S. Greiner, B. Bošković, M. Mernik, and V. Žumer, "Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 646–657, 2006.

[42] F. Neri and N. Khan, "Two local search components that move along the axes for memetic computing frameworks," in *Proceedings of the IEEE Symposium Series on Computational Intelligence*, 2014, pp. 62–69.